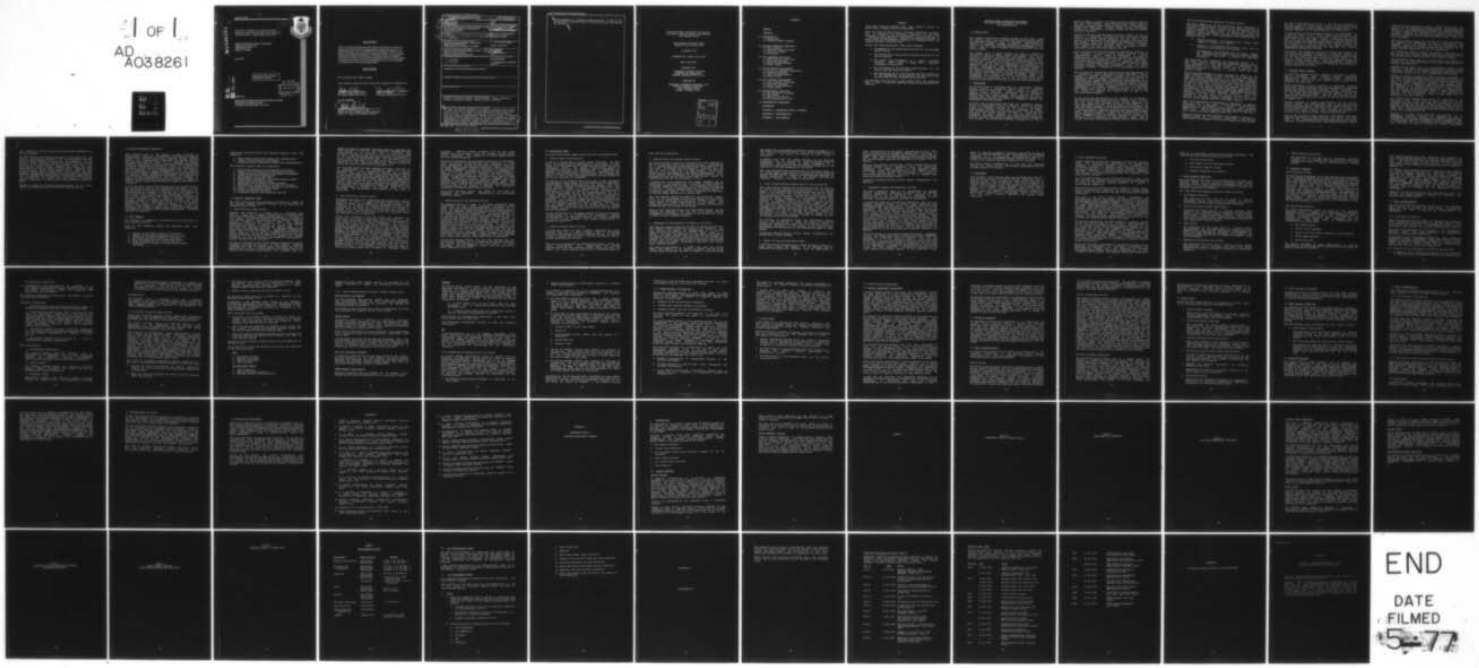
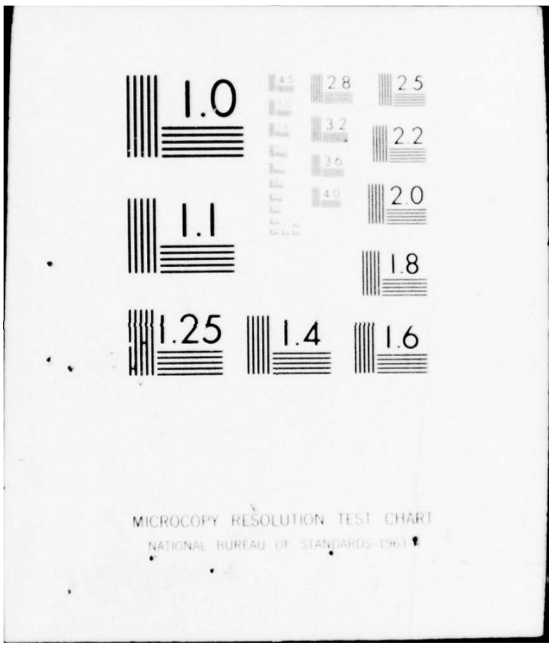
AD-A038 261 HONEYWELL INFORMATION SYSTEMS INC MCLEAN VA FEDERAL --ETC F/G 9/2
SECURITY KERNEL EVALUATION FOR MULTICS AND SECURE MULTICS DESIG--ETC(U)
AUG 76 N ADLEMAN, J R GILSON, R J SESTAK F19628-74-C-0193

UNCLASSIFIED ESD-TR-76-298 NL

1 OF 1
AD A038261



END
DATE FILMED
5-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963

ESD-TR-76-298

SECURITY KERNEL EVALUATION FOR MULTICS AND SECURE MULTICS DESIGN, DEVELOPMENT AND CERTIFICATION

Honeywell Information Systems, Incorporated
Federal Systems Operations
7900 Westpark Drive
McLean, VA 22101

August 1976

Prepared for

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS
ELECTRONIC SYSTEMS DIVISION
HANSCOM AIR FORCE BASE, MA 01731

LEGAL NOTICE

When U. S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

"This technical report has been reviewed and is approved for publication."

William R. Price

WILLIAM R. PRICE, Capt, USAF
Techniques Engineering Division

Donald P. Eriksen

DONALD P. ERIKSEN
Techniques Engineering Division

FOR THE COMMANDER

Stanley P. Dereska

STANLEY P. DERESKA, Colonel, USAF
Deputy Director, Computer Systems Engineering
Deputy for Command and Management Systems

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: ESD-TR-76-298	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): SECURITY KERNEL EVALUATION FOR MULTICS AND SECURE MULTICS DESIGN, DEVELOPMENT AND CERTIFICATION.	5. TYPE OF REPORT & PERIOD COVERED: Semi-Annual Progress Report. 1 January 1976 30 June 1976, / 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): N./Adleman, J. R./Gilson, R. J./Sestak, R. J./Ziller	8. CONTRACT OR GRANT NUMBER(s): F19628-74-C-0193
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Honeywell Information Systems, Incorporated, Federal Systems Operations, 7900 Westpark Drive, McLean, VA 22101	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: CDRL Item A022
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Command and Management Systems, Electronic Systems Division, Hanscom Air Force Base, MA 01731	12. REPORT DATE: August 1976 / 13. NUMBER OF PAGES: 61
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 67p.	15. SECURITY CLASS. (of this report): UNCLASSIFIED / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

security, security kernel, certification, kernel, operating system, multilevel access, access control, Multics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The goal of Project Guardian is to design, develop, and certify a secure Multics to provide a certified secure multilevel computer utility. The report covers activities from January to June 1976 with an introductory summary of prior work. Activities reported include simplification of the Multics operating system, development of the Multics security kernel, design of the Secure Communications Processor (SCOMP) hardware, development of the SCOMP security kernel, methodology for certifying software,

over

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

409690

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

and development of ruggedized SCOMP hardware. A list of all documentation produced under the contract during this period is included.

SECURITY KERNEL EVALUATION FOR MULTICS AND SECURE MULTICS DESIGN, DEVELOPMENT AND CERTIFICATION

SEMI-ANNUAL PROGRESS REPORT
January 1976 to June 1976

6 AUGUST 1976

CONTRACT NO. F19628-74-C-0193

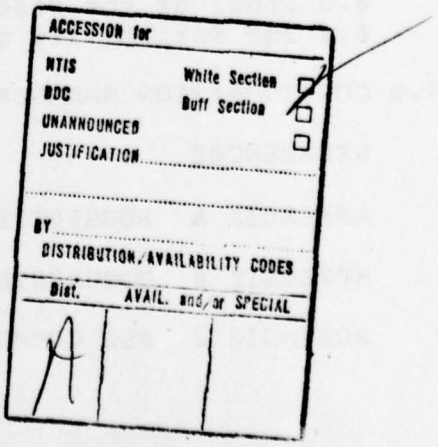
CDRL ITEM A022

PREPARED FOR

ELECTRONIC SYSTEMS DIVISION
HANSCOM AIR FORCE BASE
BEDFORD, MASSACHUSETTS 01731

PREPARED BY

HONEYWELL INFORMATION SYSTEMS, INC.
FEDERAL SYSTEMS OPERATIONS
7900 WESTPARK DRIVE
McLEAN, VIRGINIA 22101



CONTENTS

PREFACE

This report describes progress under this contract during the period from 1 January 1976 to 30 June 1976.

Under the terms of this contract, Honeywell Information Systems, Inc. is providing the technical integration of the tasks necessary to begin the design, development, and certification of a Prototype Secure Multics. Honeywell has subcontracted some of these tasks with Massachusetts Institute of Technology (MIT) and Stanford Research Institute (SRI).

During this reporting period, these tasks included:

1. Development of the system specification for the Prototype Secure Multics.

2. The restructure of the software-related functions of the current Multics.

3. The design and development of a Secure Front-End Processor (SFEP) which is based upon a securable minicomputer architecture.

4. The development of the top-level specifications for the Multics and SFEP security kernels.

5. The preparation of the certification plan and definition of the methodology to be used for the verification and certification of the Prototype Secure Multics.

MIT personnel were involved in Task 2 above while SRI performed Task 5 above during this reporting period as subcontractors to Honeywell.

SECURITY KERNEL EVALUATION FOR MULTICS AND SECURE MULTICS DESIGN, DEVELOPMENT AND CERTIFICATION

1.0 INTRODUCTION

The problem of security in computer systems has been under study for several years. The Air Force has sponsored several studies and development projects aimed at improving understanding of security in computer systems, developing a sound theoretical basis for further work, and demonstrating accomplishments in the field. Many of these projects have been associated with the Multics system.

The overall goal of these efforts has been to develop a certifiably secure computer system for general use by the military to meet their operational requirements. This report describes progress under this contract during the period from 1 January 1976 to 30 June 1976 as part of a long-term plan to take the Multics system from its present form to a prototype Multics system which can be used to demonstrate the feasibility of software certification. Three activities are being conducted in parallel to implement this plan. The first parallel effort is the design of a Multics security kernel and the simplification of the Multics operating system. The second parallel effort is the development of a Secure Front-End Processor (SFEP) for integration and certification. The third parallel effort is the development of a technology and a set of tools which allow eventual certification of the Multics and SFEP software kernels.

1.1 Background

The military faces an increasing need for operational computer systems capable of processing several levels of classified information at the same time. Present systems are unable to support secure multilevel processing due to fundamental weaknesses in their basic design, since security was not a concern when they were developed. The weakness is that current hardware/software systems are unable to adequately protect the information that they process.

Currently, the military meets the need for processing several levels of information by one of two methods. Either all security levels are processed together at the level of the highest classification present, or each level is processed by itself. Both methods have been less than satisfactory. The problem with processing all levels together is that all users and all equipment, including terminals and communications facilities, must be cleared to the highest classification that the system can ever process. The problem with separate processing is that a

separate computer system or a separate period of time is required for each level handled. Also, sharing of data between users of different clearance levels cannot be permitted. Either method is costly and inefficient. Neither method allows simultaneous handling of information at several levels for users of several levels of clearance.

Multics is the most advanced general utility system as far as security is concerned. Security was one of the initial design goals of the Multics system designers and has been a major concern of the designers and developers throughout the history of the system. Even with this concern for security, the present Multics system cannot be certified secure. Multics, however, does present the best available base upon which to build a certifiably secure multilevel computer utility.

Secure communications has also presented operational problems to the military. A secure on-line system requires a secure communications network. While the techniques of securing communications lines and terminals have been well developed, a certifiably secure communications processor is still undeveloped. A secure multilevel system must have a compatible and secure Front-End Communications Processor to be able to properly handle multiple levels of classified information. Thus, the Secure Front-End Processor is essential to the development of a secure Multics.

Both economic and operational considerations make development of a certifiably secure multilevel system desirable. Recent advances in computer technology indicate that it should be possible to produce a system that can process an arbitrary mix of classified and unclassified information simultaneously on a single computer system. The system should serve both cleared and uncleared users and should rely on the computer system's internal hardware/software controls to enforce security and need-to-know requirements. Of primary importance is that the system be certifiably secure. That is, it must be possible to prove that the system is complete and without flaw in any of its security-related aspects.

The Air Force has been working on the problem of providing a certifiably secure multilevel system for several years. In 1970, the Air Force Data Services Center (AFDSC) requested the Electronic Systems Division (ESD) to support development of an open multilevel system for the AFDSC Honeywell 635 systems. The resulting studies pointed out the severity of the problem and led to the formation of a computer security technology planning study panel. The panel's report (1) described the fundamental problems and delineated a program to develop the desired system. The panel recommended that the technical approach to the problem be "to start with a statement of an ideal system, a model, and to refine and move the statement through various levels of design

into the mechanism that implement the model system".

The basic component of the ideal system was also identified by this panel. This component is known as the Reference Monitor, an abstract mechanism that controls access of subjects (active system elements) to objects (units of information) within the computer system and enforces the rules of the military security system on such access. Three requirements were recognized for a Reference Monitor:

a. Complete Mediation - the mechanism must mediate every access of a subject to an object.

b. Isolation - the mechanism and its data bases must be protected from unauthorized alteration.

c. Verifiability - the mechanism must be small, simple, and understandable so that it can be completely tested and verified (certified) to perform its functions correctly.

The mechanism that implements the Reference Monitor in a particular computer system has been termed the security kernel. Much subsequent work has been devoted to identifying the characteristics of a security kernel and to exploring the technology involved in producing a security kernel for some computer system.

ESD initiated development of formal mathematical models of the ideal Reference Monitor in 1972. This work (2,3) resulted in a model of a secure computer system as a finite-state mechanism that makes explicit transitions from one secure state to another. The rules of the model formally define the conditions under which a transition from state to state can occur. The rules have been proven to allow only transitions that preserve the security of information in the system. The model specifies requirements for the operation of a security kernel. These requirements were taken directly from the Defense Department regulations on handling sensitive information (DoD Directive 5200.1-R). With the availability of the model, the problem of validation is now reduced to providing complete assurance that a particular security kernel behaves exactly as the model requires.

Work on the technology of certification progressed in parallel with the work on the model. In 1973, Price (4) identified a methodology for verification of a kernel. More detailed developments of this validation methodology have been reported by MITRE (5,6). Another approach has been explored which may be more suitable to large software modules (7).

Other activities have been devoted to the problem of building a security kernel for a practical system (8,9). This work has demonstrated the soundness of the basic concepts and also pointed

out some of the problems that lie in the way of realizing a security kernel on a large system. This work has been the basis for development of a secure communications processor which is an integral component of the long-range goals of this program.

A major project in the development process is the development of a security kernel for a large resource sharing system. The system chosen for this effort is Multics. There are two reasons that this choice was made. First, the hardware base of the Multics system, the Honeywell 68/80 computer, has been identified as best suited of all off-the-shelf large computer systems for the support of a security kernel (10). Second, the Multics system architecture was conceived and developed with security requirements specifically in mind.

One project, now completed, involved the design and production of a Multics system capable of supporting a two-level (Secret and Top Secret) environment for the Air Force Data Services Center (11,12). This system implements security controls based on the military access rules, but it does not completely handle the threat of a hostile penetration. From these efforts, additional insight was gained in the problems of designing and developing a security kernel for Multics.

1.2 Progress Report Summary

As a basis for program activity, Honeywell submitted a revised overall development plan, "Multics Security Integration Requirements, 1 January 1976 - 31 December 1980". Significant progress was made during this report period relative to the objectives of this plan.

System development activity resulted in functional interface definitions allowing specification and development of compatible Multics and Secure Front-End Processor (SFEP) subsystems. A significant determination was that communications will be handled by the SFEP and peripherals will be handled through the Input/Output Multiplexer (IOM). Required IOM hardware design modifications were defined in detail in a revision to the Multics I/O Report (13).

Progress continued on formal specifications for the Multics Security Kernel and simplification and reorganization of the Multics operating system. Important system issues; the impact of the DoD Integrity Policy on the user, and need to certify the discretionary access control, were resolved and included in a revision to the Security Kernel Top-Level Specification.

Most of the original or design portions of the Multics supervisor reduction tasks have now been completed. The results are being used in the continuing design and documentation of the Multics software. Several of these tasks, Name Space Management, Dynamic

Linking and Fast Processing in Ring 0, yielded results which are assisting in the achievement of program goals. Results have been applied to the standard Multics product with the implementation of design concepts being contained in the later Multics software releases. Results of the remaining supervisor reduction tasks will be factored into the Multics architecture study within the next six months.

The Multics Security Kernel Certification Plan (14) which defines the approach and methodology of kernel certification was expanded. This plan outlines how formal verification of the security properties of the security kernel with respect to the kernel Top-Level Specification will be accomplished.

The methodology employs a careful hierarchical decomposition of the design with formally stated specifications for each system function and formal assertions about each desired property.

The methodology to be used for certification was applied to several of the functions defined in the Multics Security Kernel Top-Level Specification.

Examples of the formal proof of correspondence between these functions in the Top-Level Specification and the desired security properties defined in the security model were completed to show methodology feasibility for the correspondence proof process.

Several key design decisions were made to improve performance on the SFEP. The most significant was the decision to partition the SPM into two elements. The logic area with greatest performance effect is memory management; therefore, the virtual memory management logic, including the associative cache implementation, was designed as a "daughter board" extension to the SFEP CPU. The remainder of the SPM logic was placed on a plug-in board. The SPM design was sufficiently defined to allow completion of the final Detailed Specification, Part I. SFEP documentation required for the Preliminary Design Review (PDR) will be available in the next quarter.

The remaining SFEP hardware design effort was completed to the functional level. Since many of these tasks, such as the Interface Unit design and the hardware verification, will be deferred during the next year of the program, a good functional baseline is required so these tasks can be resumed easily at a later date. This baseline was established and will be reviewed during the forthcoming SFEP PDR.

Appendix A to this report describes the progress of an independent Honeywell program to ruggedize the Level 6 minicomputer for military and heavy-duty commercial applications. Progress on this program is important in that a ruggedized Level 6 minicomputer will be the hardware base of the SFEP prototype unit. The SFEP unit which will conform to TEMPEST requirements

will contain the Security Protection Module under development in Project Guardian.

The SFEP software development progressed in conjunction with the Multics restructuring activity. An initial draft of the SFEP Security Kernel Top-Level Specification was prepared. As the security kernel primitive design was further defined, several revisions to the SFEP top-level specification were made. The SFEP security kernel design concepts are similar to those contained in the Multics security kernel design. Therefore, many of the technical issues which appear to be common to both security kernels can be resolved in a similar manner. The major technical issues which remain before the top-level specification can be completed are the approach to argument validation during ring crossings and the impact upon the kernel of including network connections to the Prototype Secure Multics through the SFEP.

Appendix B identifies the documentation prepared for Air Force review and comment during this six month reporting period.

2.0 MULTICS SUPERVISOR REDUCTION

This research phase of the program is being performed by Massachusetts Institute of Technology's Laboratory for Computer Science (LCS - formerly Project MAC) Computer Systems Research Division as a subcontractor to Honeywell. The specific goals of this continuing research effort are to identify the minimum mechanism that must be correct to guarantee computer enforcement of desired constraints on information access, to simplify the structure of that minimal mechanism to facilitate certification, and to demonstrate by test implementation that the security kernel so developed is capable of supporting all the functions of the Multics system. Because Multics permits the direct sharing of information among simultaneous computations, this research can lead to a better understanding of the structures necessary to support the primary Multics functions and, therefore, leads to a Multics system whose security features inspire a high degree of confidence.

At the conclusion of this reporting period, this research project has run for about three of its intended four year span. So far, the reductions in size and the simplification in structure of the security-sensitive software in Multics that were expected to result from the early tasks is showing significant progress. As the results from these initial tasks are becoming available and are being assimilated into the readily available version of Multics, further detailed research is emerging and being contemplated from the experience gained in the early work. Specifically, during this reporting period, a new model of process synchronization, called the "eventcount" model, was developed that leads to simpler process coordination algorithms and minimizes unnecessary inter-process communication - a feature important to security.

2.1 Task Summary

The following is a summary of the status of the various tasks in this research activity.

Prior to this reporting period, the following tasks were completed:

1. Removal of the Dynamic Linker from Ring Zero
2. Removal of Name Space Management from Ring Zero
3. Development of Fast Processes in Ring Zero
4. High Level Description of System Functionality
5. Study of Removal of User I/O from Ring Zero
6. Formulation of Criteria to Include Modules Within the Kernel

During this reporting period, the following research tasks were completed:

1. Page Control Restructure (design and implementation)
2. Traffic Control Restructure (design)
3. Answering Service Restructure (design and implementation)

The following research tasks are continuing:

1. Page Control Restructure (performance evaluation)
2. Traffic Control Restructure (performance evaluation)
3. Answering Service Restructure (performance evaluation)
4. System Initialization Restructure
5. Multi-Tasking in the User Ring
6. Methodology of Designing a Certified Computer System
7. Study of Multics Security Holes
8. Restructure of the Network Control Program
9. Management of Multiplexed I/O Streams in the Kernel
10. Study of Relationship between Reliability and Security
11. Support of User-Defined Object Types
12. Multics Performance Benchmark
13. Independent Domains and Breakproof Services

2.2 Recently Completed Tasks

The tasks listed below were completed in the past six months by the Computer Systems Research Division of the Laboratory for Computer Science at MIT.

1. Restructuring of Page Control

Research was completed on various ways to reorganize Page Control. Using the language devised under the completed task "High Level Description of System Functionality", a version of Page Control was constructed which handled read-write sequences in a separate process. This approach was then further refined to produce a version of Page Control which uses separate asynchronous processes to execute all of the Page Control functions except the act of fetching the missing page. It is felt that by isolating functions in separate processes, and constraining them by restricting the interprocess communication paths, that it will be easier to understand and certify the overall algorithm. One of the other benefits of structuring Page Control in this way is that it should be possible for several processors to take and handle a page exception simultaneously, without interfering with each other.

The goal of this task is to utilize several asynchronous parallel processes to perform the functions of Page Control. Separate processes are used to remove pages from memory and from the paging device so that a free storage pool will always exist to be used for the servicing of page faults. The processes used are

examples of the fast processes developed under the completed task "Fast Processes in Ring 0". Use of parallel processes provides simplification of the algorithm, since it eliminates some artificial interactions that occur if the functions are performed as part of the same process and which constrain the functions to run in a particular synchronized order. The new method will also scale up more effectively to a larger system since it eliminates contention on the global page table lock. During the past six months the design of this task was published (15).

Only one step remains to fully complete this task. This step is an investigation of the performance aspects of the new implementation. Initial comparisons between the standard, currently operational Page Control and this experimental version of Page Control suggest that the experimental version requires about one and one half the standard time to process a page fault. It seems that this increase in time results from the experimental version being coded in PL/I and the use of a large number of external subroutine calls which introduced considerable execution time overhead. The true magnitude of these two differences will be investigated to discover the intrinsic costs of the two algorithms. The performance investigation of this task is continuing.

2. Restructuring of Traffic Control

Techniques were explored to restructure and simplify the traffic controller in order to speed up the act of switching from one process to another and to simplify the mechanisms involved. The intention is to split the traffic controller into two parts, separating out the actual act of switching from one process to another from the more complex act of deciding which process is eligible to run. The division into policy and mechanism makes the algorithm easier to understand.

During this reporting period the design was completed to restructure the traffic controller into two levels (16). The lower level multiplexes the real processors of the system among a fixed number of so called virtual processors. By fixing in advance the number of such virtual processors, this low level processor multiplexer need make no use of the systems virtual memory facilities. Thus there is a strict isolation and ordering between the multiplexer and the virtual memory. A higher level scheduler multiplexes some of the virtual processors among all of the currently operating real Multics processes. This higher level scheduler can use all of the facilities of the Multics virtual memory, since they are implemented at a lower level. It is expected that this restructuring will clarify the relationship between Traffic Control and Page Control, and also aid in separating the idea of interprocess signalling from the idea of Traffic Control. In this proposal, no ring 0 data base (such as the current message table) will be needed for messages between

processes. Messages between processes will be sent using segments that are protected using the standard system access control mechanisms. This appears to be a great simplification over the current mechanism.

Detailed implementation and performance evaluation is proceeding. First, the low level processors using real processors is being fully implemented. Second, the high level scheduler, which will multiplex these virtual processors among real Multics processes is being implemented. Third, all portions of the system, other than Traffic Control, which must be modified or redesigned in order to run the new Traffic Controller have been identified. Included are modifications to interrupt and fault handling, changes to page fault handling, and various other smaller system changes. Recoding is in process. Fourth, a proposal has been prepared to eliminate from the system the traffic control data base known as the Interprocess Transmission Table (ITT). This removal would simplify the traffic controller significantly and be another step in the attempt to simplify the interprocess communication mechanism being used in Multics.

During this reporting period, the design of this task was completed and documented. Implementation and performance evaluation remains.

3. *Restructuring* of the Answering Service

The answering service is made up of those algorithms which authenticate the user, create processes, and manage teletype lines. This is a large interconnected set of functions, all of which are security sensitive given the current modularization. During this reporting period, the design for the rearrangement of the answering service was completed 17). The design approach is to achieve an isolation of those particular components that are in fact crucial to assure secure operation of the system. A similarity has been recognized between the creation of a new process and the entering of a new protection domain. This allows access control lists to be used to regulate the creation of processes on the behalf of any particular user. In general, this scheme avoids the need for certified software by providing the means to assure the user that a process created with the user's identification will start executing only in certain specified programs that the user provides. These programs are provided with tools which allow them to determine that the process has been brought into execution under appropriate circumstances.

During this reporting period, all the code required for the experimental redesign of the Answering Service was completed. The testing and evaluation of the new version is still proceeding. Current schedules project that this task will be fully completed in late 1976.

2.3 Continuing Tasks

The tasks listed below remain active and will be pursued by MIT.

1. Multics System Initialization

If one is to certify that a system works correctly, one must begin by verifying the "initial state" of that system. For this reason it is very important to understand how the Multics system initializes itself. The current initialization system is relatively unstructured and confusing and is apparently not amenable to verification or certification. A proposal has been made for restructuring system initialization that reduces the amount of code in the initialization phase and that simplifies the task of verifying the remainder.

The approach is to recognize that much of what is now considered initialization ought rather to be considered as reconfiguration. Initialization is then decomposed into two phases. The first phase involves getting a minimal Multics up and running. The second phase is a series of reconfigurations based on input describing the actual configuration in use. The advantage of this strategy is that all of the reconfigurations run in a complete, operational Multics environment, which is much easier to understand than a partial and ever changing environment as presented in the various stages of the current initialization procedure. Also, since the minimal Multics is independent of the actual configuration (given the minimum hardware required), it can be largely generated as the system tape is created. Algorithms which run at the time that the system tape is created are easier to verify since they too run in a fully operational Multics environment.

Initial verification of the design approach of structuring system initialization as a collection of sequential, dynamic reconfigurations of appropriate hardware and software subsystems is currently being verified. Final design documentation is expected in early 1977.

2. Study of Multics Security Holes

A listing and report is made annually cataloging all known security flaws in the Multics system, ways to violate the security of the system, or ways to crash the system. An attempt is made to analyze each flaw and to identify the general class of problem represented by the flaw.

This task is a continuing task requiring an on-going review and annual documentation. This documentation has limited distribution by its very sensitive nature. During the past six months, the annual report on this task for the period ending 30

June 1975 was distributed.

3. Restructuring the Network Control Program

The Network Control Program has represented an ideal candidate to use in an experiment involving a multiple process implementation of a control algorithm, as discussed under "Multitasking in the User Ring". The Network Control Program is concerned with the flow of data to and from the ARPA Network. Its principle function is the management of a multiplexed communications path, which implies the management of multiplexed buffers. It was proposed that the flow of data between these various buffers be implemented using the fast processes in ring 0.

A related area is the possibility that common elements may be identified in the software that is required to handle different multiplexed communication streams. It is possible that there is a similar function in the software that interfaces the ARPANET and the 355. A buffer manager routine is an obvious example of a possible common routine. It is very interesting and profitable to identify these modules and to isolate them.

This task markedly reduced the amount of code within the kernel. Also, if all multiplexed communication streams could be handled by one set of kernel modules, such as the network server, this could result in great simplification of the kernel. This observation also has led to the start of a general re-examination of the way that I/O is done by the system.

Early in this reporting period, the restructured Network Control Program was installed. The task was then merged with the "Management of Multiplexed I/O Streams in the Kernel" task below to better utilize available manpower.

4. Management of Multiplexed I/O Streams in the Kernel

This task was formerly known as the "New I/O Buffer Strategy" but was renamed to better reflect the intent of the research. This task involves the design and implementation of a new I/O buffering strategy which uses the virtual memory itself as a buffer. The task has become part of the task of restructuring the Network Control Program. The task arose from an attempt to increase the efficiency of transmitting data to and from the ARPA Network and at the same time to gain a more basic understanding of the interaction between Input/Output functions and virtual memory computer systems.

The result of this design is a buffer that uses the virtual memory and appears to be infinite in length. The use of the virtual memory eliminates any need to compact or otherwise manage

the buffer area, thus reducing overhead. Since the buffer is in the process virtual address space, it is directly accessible to a user process. This avoids the copying of data to make it accessible.

It appears that this I/O buffer strategy can be exploited successfully for all devices for which the system nucleus is responsible. This unification of buffer management is a significant contribution to the certification project due to its reduction of bulk and complexity in the kernel.

This task had previously progressed from the conceptual phase to the design phase in that tape I/O services were being considered through the ARPA Network interface to determine the magnitude of thruput degradation. A study of the DN-6600 protocols was completed. Progress on this task is continuing.

5. Study of Relationships between Security and Reliability

This task involves studying the relationship between the security of a system and the reliability of that system. In the Multics system, it is presumed that a system failure may have an unknown effect on the security status of the system; this is the reason that the system is shut down, salvaged, and restarted after every unexplained system failure. It is not obvious, however, that security is directly dependent on reliability. If it were possible to determine that certain classes of computer failure could not influence the security state of the system, then the two functions would have nothing to do with each other. More strongly, it is possible that a highly reliable system contains mechanisms that are not desirable from the viewpoint of security. For example, one way to increase the reliable storage of information on a system is to make several copies of that information; many copies, however, increase the probability that the information may be compromised.

In an attempt to understand better the relationship between security and reliability, a study has begun of a variety of systems that maintain high reliability as one of their goals, to investigate mechanisms in the system in the light of their implications for the security of the information stored on that system.

During this reporting period, initial design documentation was revised and is being reviewed.

6. Support of User Defined Object Types

A directory can be considered as an object defined in terms of a lower level object, the segment. It is possible that the mechanisms that define the directory could be generalized to

allow the definition of new object types defined by users. This sort of ability has been provided in systems that are based on capabilities, but not in systems that are based on access control lists. This task has been considering the question of whether user defined object types can be supported in a system such as Multics.

There are two projects in this area. First, consideration of how extended objects can be supported using an appropriate combination of access control list and capability based mechanisms. Second, the implementation of user defined objects in a pure access control list environment. This project is considering what policies for protection can be imposed upon these user defined extended objects.

During this reporting period, initial design documentation was prepared.

7. Independent Domains and Breakproof Services

This task is exploring some of the implications of removing certain traditional supervisor functions from the Multics security kernel and is exploring an extension of the functionality of the Multics protection mechanisms to allow multiple, independent domains to be part of one process.

A traditional supervisor includes many mechanisms that are not security sensitive simply to protect these mechanisms from accidental damage from user errors. To produce a security kernel for Multics, many such mechanisms are being moved out of the supervisor. By moving them to the user environment, mechanisms such as the linker, the reference name manager, and the search rules become breakable, which could make the system harder to use. Fortunately, the Multics protection rings provide a place to protect non-kernel mechanisms that should be breakproof. They can execute in a ring, say ring 3, above the kernel but below the normal user ring. Because all data bases managed by these service mechanisms are private to a process, they are not part of the security kernel and need not be certified, yet they cannot be broken inadvertently by user errors. Part of this project is the determination as to how to provide such breakproof services for Multics.

The second aspect of this project concerns protected subsystems. Multics has always supported user-defined protected subsystems, although the protection rings can provide only one way protection. It is not possible, however, to use the rings to protect both subsystems and breakproof services at the same time in the same process without making the breakproof services common to all subsystems in a process and therefore part of the security kernel for those subsystems. The essential difficulty is the total ordering of privilege implied by the protection rings.

Thus, to provide breakproof services, some other way must be found to protect subsystems, if the functionality of protected subsystems is to be maintained. The method being explored is simulating multiple independent domains (containing rings) in a process using multiple descriptor segments for a process.

Preliminary design documentation of this task was completed during this report period. Experimental implementation of the proposed design is continuing.

2.4 Conclusion

This section of this progress report has presented the tasks of a large research project to evolve the Multics supervisor into a security kernel which is capable of supporting the functions of Multics completely and efficiently. The broad objective is finding ways to reduce the size and complexity of the software by using a multi-pronged attack approach. This software must be correct for a shared general-purpose system to be secure. Reduced size and complexity of security-relevant software is a prerequisite to performing a convincingly logical verification that the system correctly implements the claimed access constraints.

3.0 SFEP HARDWARE ACTIVITIES

Current computers utilized for Information Storage and Retrieval (IS&R) and Communications applications are fundamentally incapable of providing adequate security protection for concurrent processing of DoD multilevel classified information. The current approach of providing either physical separation (multiple facilities) or temporal separation (with intermediate sanitization), is prohibitively costly for the typically large scale systems required.

The Front-End Processors to be utilized with secure large scale IS&R systems and computer utilities require multilevel security in order to extend the security perimeter to include a communications capability.

The solution to this problem is based on security design concepts which are demonstrably sufficient to effect a secure system design which is certifiable and does not prohibitively degrade performance.

This part of the Secure Multics Design, Development and Certification Program encompasses the initiation of the design and development and prototype fabrication plans for a militarized Secure Communications Processor (SCOMP). A specific application has been identified which will demonstrate the functionality of the SCOMP. This application is to function as a Secure Front-End Processor (SFEP) for a prototype secure machine of the Honeywell Series 60 Level 68 Multics class in a communications and network environment. The SFEP will accordingly include the interface unit to the host machine. Also included is initiation of hardware verification plans, development of militarization plans and plans for communications network interfaces.

The security approach for the SFEP is based on concepts derived during the previous phase of this contract. These concepts are in turn based on the Reference Monitor concept developed by the Air Force (and others) (2) which implements the access rules and algorithms of the Bell and LaPadula math-model (3). The math-model in turn models the access rules of the DoD Information Security System. The math-model is a representation of finite discrete-state mechanisms wherein state-transitions are explicitly governed by rules of the model. Since the math-model is a discrete-state model, it is correlatable to digital computer system architectures.

While the functional requirements of the Reference Monitor may be performed interpretively (in software), performance and certifiability require that the functions be carefully distributed among hardware and software implementation in order to effect a useful system. The computer architecture selected is

based on the established isolation and mediation mechanisms. The following specific hardware features are noted:

1. Virtual memory system

2. High speed cache for descriptor storage

3. Hierarchical domains (rings)

4. Hardware supported ring crossing

3.1 SFEP Hardware Objectives

The objectives of this phase of the program were to initiate the detailed design of the Security Protection Module (SPM), determine suitable design verification methodology, and plan for fabrication of the development unit of an SFEP for use with the Honeywell 6000/Series 60 computers.

The SFEP effort encompassed five major tasks as follows:

1. Minicomputer Selection

   The objective of this task was to select a suitable commercial computer base which is seurable and can be militarized, for the range of applications delineated.

2. Security Protection Module (SPM) Design

   The objective of this task was to initiate detailed logic design of an SPM which can be integrated into the selected computer base to add the necessary security controls which will effect a useful secure computer system. This included performing many design level trade studies which impacted the logic design and circuitry layout.

3. SCOMP Design

   The objectives of this task were to: develop initial specifications for the SFEP subsystem, including functional configuration, interfaces and performance requirements; to carry out SFEP performance analysis; and establish the interface between the SPM and the commercial Honeywell Level 6/40 minicomputer.

4. 6000/Series 60 Interface Unit Design

   The objective of this task was to develop initial design specifications for an interface unit which will permit interfacing the SCOMP to Honeywell 6000/Series 60 large-scale computers.

5. SCOMP Hardware Verification

The objective of this task was to investigate techniques suitable for (a) hardware verification; and (b) deriving a probabilistic measure of security compromise due to hardware failure.

3.2 Technical Approach

SFEP Functional Design

The SFEP Security approach was to base SFEP Security requirements on the concepts delineated in the Architecture Study Final Report (18). The approach for selection of the minicomputers and the preliminary designs for both the SPM and 6000/Series 60 Interface Unit (IU) were to base them on Secure Communications Processor Functional Specification, which defined requirements for a Front-End Processor as well as a stand alone communications processor and a network interface processor. The approach to deriving the ultimately recommended specific implementations of the SPM and 6000/Series 60 IU was then based on the selected minicomputer and numerous design tradeoffs conducted under Air Force guidance. Detailed functional specifications (Detail Specifications, Part I documents) were then developed for both the SPM and 6000/Series 60 IU.

SFEP Environmental Design

The SFEP Environmental Design is based on the Honeywell Ruggedized Level 6 computer now in development at Honeywell's Aerospace Division (See Appendix A). This ruggedized Level 6 computer is functionally based on and is compatible with Honeywell Information System's (HIS) commercial line of minicomputers. Five major areas of modifications for ruggedization are listed below:

1. New chassis design
2. Circuit card stiffening
3. Option of pin and socket connectors at bus interface
4. Power supply mounting
5. New control panel

The designs developed for these modifications are directly applicable to the additional SPM and 6000/Series 60 IU boards and modules required for the SFEP.

The rugged minicomputer has been fabricated and assembled and will be qualified by Honeywell to its Detail Specifications (DS), Part 1. The qualification program should be completed during the next three month phase of the program. A more detailed description of the ruggedized minicomputer program is included as Appendix A.

The TEMPEST Design approach developed control plans that provide the design guidance necessary for compliance with RED/BLACK separation and TEMPEST compatibility. The power filtering as well as the chassis shielding have been designed to meet the conduction and radiation requirements of the TEMPEST specifications. The qualification program also includes the TEMPEST testing. The task to complete the isolation strippers for the RED/BLACK separation has been deferred until the prototype phase of the Program when the Secure Communications Processor will be installed into the operational prototype environment.

Similarly, the EMC design approach developed control plans that provide the design guidance necessary for compliance with MIL-STD-461A and MIL-STD-462.

3.3 Major Accomplishments

The following paragraphs describe the details of the individual tasks required to initiate the SFEP design. The technical approach to the tasks is described and accomplishments are delineated.

3.3.1 Minicomputer Selection

During the previous six month phase an analysis was performed which defined the criteria by which a trade study could be done. This criteria was in the form of requirements placed on the minicomputer base required for a secure communications processor.

This task is now completed with the update of the Minicomputer Selection Report (19) that discusses the requirements, candidates, and trade-off methodology.

The Honeywell Level 6 minicomputer series with their respective ruggedized military counterparts (RL6) were selected as the computers which best satisfy the requirements of this program. Within the Level 6 family, Level 6/40 was selected as the hardware base for the initial SFEP application.

The selection of the minicomputer was done on the basis of:

A. A comparison of the architectural features of the candidates to those desirable architectural features of a secure system

as determined by MITRE (20).

B. A comparison of the capabilities of the candidates to the requirements derived from the program Statement of Work (21), the SCOMP Architectural Study, and potential SCOMP applications such as SATIN IV.

The top level requirements affecting the minicomputer selection are summarized as follows:

Functional Requirements

1. The architecture should be bus-structured to support an essentially autonomous Security Protection Module (SPM).

2. The architecture should have basic functionality such that the minicomputer, supported by the SPM, will provide the functionality required for effective implementation of a Reference Monitor. This includes multiple machine states, fast context switching, suitable address space definition, continuous access checks, I/O access control, variable access permissions, real-time support and interprocess communication features.

3. The selected computer must have sufficient performance to satisfy the Front-End Processor application requirements for the Honeywell 6000/Series 60 computers including Multics.

4. The selected computer must be suitable for a range of communications processor applications.

Other Requirements

1. Environmental Requirements

The selected computer should be compatible with, or modifiable to be compatible with, a range of selected physical and electrical military environmental specifications applicable to SFEP and SCOMP applications.

2. Product Support

The selected computer should have continuing corporate product support throughout the useful life of the 6000/Series 60 family of computers.

3. Management Issues

The selected computer base should be capable of being modified to support the SPM. The availability of detailed

engineering drawings and data to Honeywell to support the integration of the minicomputer to the secure Multics system was of importance. The overall cost of the selected minicomputer with a variety of system options for varied configurations was also included in the selection criteria.

Recommendation

The Honeywell Level 6 computers with their ruggedized counterparts were selected as the computers which best satisfy the salient requirements. Within the Level 6 family, medium and upper computer models were selected as the baseline for the initial SFEP application.

3.3.2 Security Protection Module Design

During this six month reporting period, numerous SPM functional, performance, and implementation trade-offs were performed and discussed with the Air Force in Technical Interchange meetings.

The initial activity during this reporting period was the completion of the development of the Detailed Design Specification which reflected the trade-off results. The Design Specification was submitted to the Air Force (22).

Upon submittal of the Functional Design Specification, initial logic design for the SPM was started. Preliminary logic design for the SPM dictated that in order to satisfy performance objectives, the SPM would have to be partitioned into two elements, a Virtual Memory Interface Unit (VMIU), and an autonomous SPM module. The VMIU will be mounted directly on the Central Processor Unit (CPU) board as a daughterboard. The VMIU provides for descriptor storage and will perform high speed translation of virtual to real memory addresses for performance enhancement. The autonomous SPM pluggable module will contain logic which provides resource access mediation, descriptor access controls, back-up storage of descriptors, storage of I/O device controls for initializing and updating the VMIU, bus and VMIU interfaces for all microprogram control logic.

As a result of a Technical Interchange meeting conducted on 6 April 1976, the following decisions concerning the SPM were made:

A. Sixteen (16) bits of base address for indirect descriptors (Modulo 16) and 13 bits for direct descriptors (Modulo 128) is acceptable.

B. Three (3) rings are acceptable for Secure Front-End Processor (SFEP) applications.

C. The descriptor hit ratio on the VMIU must be maximized since performance will primarily be determined by the VMIU. Various implementations including associating on Normal Segment Number (NSN) should be evaluated.

D. Update DS Part I specification for Air Force review.

The decisions listed above were reflected in a revision of the SPM Detail Specification (23).

In addition, as the detailed logic design of the commercial minicomputer approached finalization, it was necessary to assure that the CPU/SPM interfaces were compatible. Accordingly, a four week engineering coordination activity was scheduled at the Boston Computer Facility.

Major objectives were as follows:

A. Investigate the CPU/VMIU connector interface and define any changes required in this interface to enable it to mutually satisfy the requirements of both the Commercial Memory Protection Unit and the VMIU of the SFEP.

B. Learn the detailed commercial minicomputer logic design for the purpose of defining the remaining interface between the CPU and the Security Protection Module (SPM).

C. Learn the firmware development process and how to perform the firmware modifications required for the SFEP at the close of this reporting period.

Resolution of all CPU/SPM interface details will be completed by the end of July.

Logic design flow charts and detailed logic design are continuing in the following areas:

VMIU

1. Descriptor storage
2. Tag and limit checks
3. Address translation
4. Fault controls

SPM Autonomous Module

1. Adder/comparator
2. Back-up descriptor storage
3. Data and control holding registers

Completion of the logic design activity is planned for the October time-frame to support the Preliminary Design Review (PDR).

3.3.3 Secure Communications Processor (SCOMP) Systems Design

Specifications and Manuals

The SFEP Subsystem Specification (Draft) has been completed. This specification defines the SFEP functional requirements, allocates functions to hardware and software, and defines the hardware/software interfaces.

A preliminary SFEP Processor Manual and the SFEP Master Test Plan were completed and submitted for Air Force review.

System Design

Performance analysis is continuing on preferred descriptor storage selection techniques for the VMIU and back-up cache. Parametric estimates of descriptor cache hit-ratios are being derived based on program traces from compiler and assembler generated commercial applications software.

Under further evaluation are a four descriptor, fully associative cache, and various sizes of descriptor pseudo associative cache for the VMIU.

A SCOMP Master Test Plan for the SFEP was completed. This test plan outlines the various tests required for the SCOMP during the Concept Development Phase: however the plan is limited to the tests to be performed prior to the delivery of the SCOMP unit to the Multics site for integration with the Multics host.

Test and Evaluation Software

An additional objective of this task was to plan for the eventual test and evaluation of the SFEP hardware utilizing software designed and developed for the purpose of test and evaluation. This test and evaluation software is described in the Test and Evaluation Software Plan, dated 13 January 1976.

SCOMP Design Militarization

The major objective under this subtask was to develop initial TEMPEST and EMC design specifications for a militarized SCOMP.

TEMPEST

The initial system design effort was the definition of the RED/BLACK requirements. A typical situation would find the SCOMP operating in a secure RED area with both high and low speed RED lines and low to medium speed BLACK lines. High speed and low speed are somewhat nebulous terms; in the current context, low speed refers typically to 100-9600 BPS and high speed refers to data rates greater than 50 KBPS. The rationale for the typical situation is as follows:

1. A high speed line is most often used for short distances; that is, beginning and ending within the same controlled area.

2. A typical remote BLACK user would communicate through a channel conforming to MIL-STD-188 or RS-232.

RED users are not precluded from BLACK data; a RED user would obtain BLACK data on a RED line.

The SCOMP design incorporated features to meet the isolation requirements.

EMC

As stated previously, all of the TEMPEST requirements except those specifically relating to the RED/BLACK isolation have been incorporated into the ruggedized version of the SCOMP. A responsible TEMPEST program must be interrelated to the EMC design. The EMC requirements are MIL-STD-461A and MIL-STD-462. The major EMC design constraint is the imposition of TEMPEST, as the two disciplines are interrelated but not necessarily compatible.

3.3.4 6000/Series 60 Interface Unit (IU) Design

An initial Design Specification dated 31 January 1976 was completed and submitted to the Air Force for review and comment. This Design Specification implemented the IU as an unmapped device requiring control to be performed by a trusted process. The unmapped approach was specified in the initial specification primarily to support a scatter/gather capability in the IU. It was subsequently determined that the support of general scatter/gather operations on the SFEP was not necessary since each IU connect will cover a data transfer for a single terminal. In addition, the unmapped approach is undesirable for the following reasons:

A. An otherwise unused access mechanism is introduced in the SFEP system.

B. Requires absolutization of SFEP memory addresses in software rather than hardware.

In an effort to implement the IU as a standard SFEP device (i.e., capable of operating either mapped or premapped), the following four approaches were evaluated:

A. All SFEP memory accessed by the IU (for a single connect) would be by a single descriptor. Both Data Control Words (DCW's) and data buffers would be required to reside within the same page (or segment). This type of structure would require the kernel to copy data from (to) user space to (from) kernel space. The DCW's would be contained in Kernel space.

B. To eliminate the copy requirement of Approach A, the sharing of data buffers among users could be disallowed once the user calls the kernel to send/receive a message. Therefore, the data buffer area would remain under kernel control until a transfer is completed. The following sequence would be required for this approach.

   1. Construct DCW's in user data buffer
   2. Connect IU
   3. Block accesses to data buffer area with kernel until termination
   4. Destroy DCW list
   5. Return to user

C. Use two IU commands (set-up and connect) to initiate IU transactions. Set-up would transfer an entire DCW list to IU storage. Connect would define the user's data address to the IU. This allows the DCW list to remain in kernel space and the data to remain in user space.

D. Use two IU commands (set-up and connect) to initiate IU transactions. The set-up command would define a DCW address pointer to the IU. Each command would be associated with different ID names in the IU which allows the SPM to simultaneously retain memory reference descriptors for DCW's (in kernel space) and data (in user space).

Evaluation of the various approaches culminated in a new Detail Specification which reflects an IU implementation according to Approach D. The new Detail Specification was submitted to the Air Force for review and comment. A preliminary IU design

investigation was performed which estimated the parts and power requirements based on a detailed block diagram.

3.3.5 SCOMP Hardware Verification

During this reporting period, a draft final report on SCOMP Hardware Verification Methodologies was completed and submitted (23). In this report, the results of trade-off studies were presented in three areas:

A. Hardware design certification techniques.

B. Probabilistic measures analysis techniques.

C. Physical product test and certification criteria.

Air Force technical comments and approval of the draft were received. The update of the report for publication is in progress.

The major technical effort in this reporting period was in the reassessment of the hardware design certification methodology trade-off in which Register Transfer Level (R.T.L.) simulation was initially recommended. Further study of the R.T.L. simulator interface characteristics revealed the need to create a link between the R.T.L. simulator and the SFEP instruction simulator if the latter was to be employed as the source of the reference monitor exercise problem. The link would need to be a dynamic software translator between the R.T.L. (H6080 code) and the SFEP instruction simulator (Multics FORTRAN). In view of this complication, it became necessary to reevaluate the earlier trade-off between R.T.L. simulation and a SFEP instruction simulator based hardware certification approach.

The evaluation resulted in a recommendation to employ the SFEP instruction simulator (also to be used in SFEP kernel development) augmented with SPM functionality program modules developed specifically for SFEP hardware verification. The major advantages of this approach over R.T.L. are:

A. Hardware certification is accomplished entirely in the Multics environment.

B. The same simulator is used for both kernel development and hardware verification.

C. It is simpler to create SPM functionality modules than to modify the R.T.L. simulator to accommodate a dynamic CPU interface link.

The technical rationale supporting the above conclusions is described in the updated Hardware Verification Methodologies final report.

A lesser level of effort was applied to refining the probabilistic measurement approach. Technical elements of this trade-off in the draft report were the subject of several Air Force comments. While the selected functional level of analysis on a baseline SFEP is established as appropriate, two major elements of the trade baseline SFEP configuration and compromise definitions are not yet satisfactorily defined. Secondary probabilistic measurement issues identified in the April, 1976 technical interchange with the Air Force were resolved and incorporated in the Final Report. Efforts to finalize security compromise definitions (for conditions resulting from hardware failure) are continuing.

3.4 Future Plans

Future plans are to complete the SFEP design; fabricate, test and evaluate prototype SFEPs and support integration into the Prototype Secure Multics systems on a schedule consistent with program requirements.

Specific efforts to complete the design, fabrication, integration and test evaluation of the SPM hardware and the Level 6 minicomputer are as follows:

A. Design, development and fabrication of Security Protection modules and a 6000/Series 60 Interface Unit to permit the integration of a modified minicomputer into a large scale system and to function as Secure Front-End Processors.

B. The fabrication of a non-ruggedized SCOMP unit to serve as a hardware and software checkout, development, and demonstration facility.

C. The fabrication of a non-ruggedized SFEP unit for Multics integration tests.

4.0 SECURE MULTICS DEVELOPMENT

4.1 Multics Development Objectives

A secure computer system is one which can successfully protect all data entrusted to it from unauthorized disclosure. This is the basic definition of system security or more specifically system software security which guides the Guardian project. Issues of physical security which can deny service to authorized users are specifically ignored here (e.g., fire, flood, etc.). The major concern is to counter all security threats which would allow someone to steal information (or data) from the computer system. The security threats of general interest fall into three logical areas: malicious persons external to the system, authorized users of the system and collusion between authorized users.

The threats from malicious persons external to the system are not particularly interesting to the system software designer. These threats include: tapping communication lines; stealing listings, tapes, terminal output or other data generated by the system; stealing passwords of authorized users; monitoring electromagnetic emanations from the hardware; or unauthorized actions by operations or administrative personnel. Each of the threats mentioned can only be countered by physical or procedural security measures external to the computer system. The only external threats of interest to the system software designer are illegal attempts to enter the system (login) and operational errors. These are solved by the use of passwords for user authentication and by providing unambiguous instructions and/or messages to operations personnel.

The remaining security threats come from users authorized to enter into and use the system. This is the area of particular interest in this development effort. The less severe internal threats of browsing by a curious user and accidental granting of access have been addressed by the implementation of the Access Isolation Mechanism. The insidious threats of a Trojan Horse program (authorized user unknowingly cooperating with an unauthorized user) or system penetration remain to be solved.

Within the Multics architecture, a general solution to the threat of a Trojan Horse has not been found. However, for a Trojan Horse program to be able to compromise data, it must be able to communicate between security levels. Therefore, one requirement of this effort is to eliminate all communication paths which would allow a program to read data of one security level and write it where it could be read from a lower security level.

A user who can penetrate the supervisory elements of the operating system may be able to invalidate all the access control mechanisms. A penetration can occur from incorrect implementation of the various protection mechanisms or from a

malicious programmer inserting special code sequences to provide a "trap door" (link from one security level to another) into the operating system. Therefore, another requirement of this effort is to verify the correct implementation of the Multics operating system (security kernel) and to verify that no trap doors exist.

The Multics protection mechanisms are implemented within the most privileged protection ring, ring 0. Unfortunately, there are a large number of programs in ring 0 which are very complex. The interactions between these programs are also complex and often subtle or obscure. In addition, there are no mechanisms to protect programs and data within ring 0 from errors in other programs in this ring. Therefore, any attempt to verify the correctness of the current Multics supervisor as it exists is doomed to failure from the start.

4.2 Technical Approach

The approach to meeting the requirements is to restructure the current Multics operating system to isolate the primitive mechanisms which implement the security access controls. This will form the reference monitor or security kernel of Multics. The mathematical model of computer security is the criterion used in defining the interface between the kernel and other parts of the system. Good engineering practice requires that the current operating system be molded into the new structure rather than attempting a complete top-down redesign. It is expected that several iterations between top-down specification for correctness proofs and bottom-up design for engineering feasibility will be needed.

4.3 Major Accomplishments

The activities over the last six months have concentrated on the top-level specification of the security kernel, external Input/Output (I/O), and the subsequent restructuring of the remaining Multics supervisor functions.

Multics Kernel

The Multics security kernel contains all functions which provide access control decisions and all hardware/software mechanisms necessary to support the access control functions. It is these functions which must eventually be certified correct for Multics to be secure. The security kernel includes all Ring 0 software (simplified, of course), all trusted processes, the Central Processing Unit (CPU) hardware itself, the memory addressing hardware, the IOM and channel hardware, internal I/O functions, the SFEP communications interface, and the external peripheral I/O interface. During this reporting period, alternative designs

for the Multics kernel were evaluated. The rationale for removal of Directory Control from the kernel was established. A detailed description of the kernel functions is being prepared in the Multics Top Level Kernel Specification for submission to the Air Force in the next reporting period.

Secure Input/Output Services

The means of providing secure internal I/O functions has caused the greatest concern to the project. The original MITRE proposal of handling all external I/O through the SFEP has been replaced due to unwieldy engineering considerations. The high bandwidth interface requirement needed to support high speed devices and the extra problems of supporting this interface over a distance of 2000 feet was determined to be less practical than our primary alternative. We have chosen to provide high speed peripheral I/O services through the IOM which will have to be slightly modified. This method of supporting I/O is presently being provided within Multics. Some hardware modifications to the IOM have been designed which will show that the IOM and the current software mechanism (ioi_) form a complete reference monitor for these I/O functions to support peripheral I/O . The SFEP is still required for handling the external communications I/O functions. It has been determined that the DATANET 6600 (the current communications processor) and the current front-end processor software (Multics Communications System - MCS) cannot be certified. Since, by its nature, a front-end processor must handle multilevel data, the front-end processor kernel must also be certified just like the Multics kernel. The SFEP approach is the only way found to support and provide the environment for certification. The results of this I/O study are being documented in a comprehensive technical report for submission to the Air Force in the next reporting period. This report will also describe the technique to model secure I/O. This technique will then be applied to test the security of the I/O design. Discussions of the hardware and software structure leading to this design will be included.

Multics Supervisor Restructure

The simplified security kernel and, to a lesser extent, the changes to handle external I/O have required the restructure of several supervisor functions. The possibility of removing the directory control function from the security kernel was investigated and is very attractive, as opposed to the approach proposed by MITRE. Some new administrative mechanisms are being proposed to support I/O device assignment according to the security model. The New Storage System (NSS) design has enabled some new system features to be defined. A most desirable feature is to allow creation of upgraded segments. This may become possible when some proposed changes to the quota mechanism are fully investigated. The changes to the user interface and

restructuring of supervisor functions are being documented in the Specification for a Prototype Secure Multics System to be initially submitted to the Air Force during the latter part of 1976, as a revision to the original submission of 31 January 1976.

4.4 Future Plans

During the next program period, it is planned to initiate effort on the following Secure Multics Development tasks:

1. System Program Language

   This task will provide support to the ongoing Honeywell system programming development effort on issues of correspondence proofs and certification.

2. Formal Programming Technology

   The technology for formal program specification is relatively new and has changed greatly during the last years. The Multics environment will put some restrictions on the choice of a formal language which can be used effectively in the correspondence proofs. Investigation in this area will be started.

3. Informal Prototype Kernel Description

   Effort will continue on the expansion of the preliminary top-level description of the prototype kernel into a more detailed functional description of modules that will form the kernel, including their interactions.

4. Preliminary Multics Kernel Module Specifications

   The task of taking the top-level specifications of the security kernel software functions and converting them into calling sequences and argument descriptions for each module will be started.

5. Redesign and Informal Description of Non-Kernel Supervisor Functions

   Modification of supervisor interfaces affected by the kernel design will be considered.

6. Performance Studies

   Effort will be initiated to establish the measurement, estimating and reporting of performance, compatibility and security of the kernel-based Multics.

5.0 SFEP SOFTWARE DEVELOPMENT

The Reference Monitor concepts require not only SFEP hardware implementation but also require a security kernel which operates at the most privileged level to ensure the correct implementation of security policy and procedures.

5.1 SFEP Software Objectives

The objectives of this phase of the program were to specify a security kernel for the Secure Communications Processor (SCOMP) and to begin the design of this security kernel. The SCOMP security kernel should be general purpose in nature and a suitable base for additional software to permit application of the SCOMP kernel in environments other than just the the Multics Secure Front-End Processor (SFEP) being developed for this program.

The SFEP software effort encompasses two major tasks as follows:

1. SFEP Security Kernel Design

   The objective of this task was to continue the detailed primitive design of the SFEP security kernel and to complete the Top Level Specification (TLS) for this kernel.

2. SFEP Operating System and Applications Software Design

   The objective of this task was to: define the functional requirements for SFEP operating system and application software and to evaluate the suitability of Honeywell Level 6 Basic Executive Software (BES) for use in the SFEP.

5.2 Technical Approach

The SCOMP security kernel approach was to satisfy the requirements outlined in the initial kernel specifications provided by the Air Force. These requirements were then reviewed in light of the Secure Communications Processor Specification prepared by Honeywell and modified to remain in accordance with the SCOMP specifications. Once the requirements were defined for a general purpose SCOMP kernel, then these requirements were translated into a SFEP Kernel Top Level Specification.

5.3 Major Accomplishments

The following paragraphs describe the details of the individual tasks required to initiate the SFEP software design.

5.3.1 SFEP Security Kernel Design

The Security Kernel's primary function is one of mediation, i.e., the enforcement of security/integrity rules, whenever an attempt to move information between objects is made. Effort under this security kernel development task has been aimed at the functional design of a general purpose security kernel (i.e., one that is applicable to a variety of applications, including front-end and communication processing applications) and the preparation of a Top Level Specification (TLS) for this kernel. The TLS was reviewed with the Air Force and initial revisions were completed.

The TLS covers the interface between untrusted processes and distributed trusted process parts of the kernel. It provides the untrusted processes access to the three types of kernel objects, i.e., process, memory, and I/O devices. The TLS includes a specification of the visible functions (or "gates") controlling the objects as well as a specification of the objects and their components (e.g., security/integrity level, ring brackets, etc.)

The TLS is written in a formal language to make it suitable for formal proofs of correctness. The present TLS is written in an interim language. Translation to SRI's "SPECIAL" language (used for the Multics kernel specification) is planned in the third quarter of 1976.

Current activity has centered on argument validation. Argument validation can be performed with the current SFEP architecture; however, substantial overhead is encountered in copying arguments at each level during inter-ring transfers. A cleaner approach to argument validation is being pursued.

5.3.2 SFEP Operating System and Application Software Design

Initial functional requirements for the SFEP Operating System and application software were completed and incorporated into the SFEP Subsystem Specification. Currently, minimal activity is centered on a modular analysis of the commercial Level 6 operating system and support software to ascertain the feasibility of utilizing modules of this software within the SFEP operating system and application software.

5.4 Future Plans

On-going SFEP software development will proceed through the following series of steps. First, formal specifications for the

security kernel will be completed, reviewed with the Air Force, and submitted to the certification process. The prototype kernel and SFEP operating system/applications software package will be implemented. This software package will be used to demonstrate a free-standing capability. Limited kernel verification or technical certification will also be carried out. This will include demonstrating correspondence between the Air Force model for a secure computer system and the formal kernel specifications and, for one module only, correspondence between the formal specification and the higher order language representation. The kernel and operating systems software will be integrated with the Multics front-end applications software and used in the integrated Multics demonstration. Finally, software that provides an interface to a network such as ARPANET will be implemented and demonstrated.

6.0 CERTIFICATION ACTIVITIES

During this reporting period, significant progresss was achieved in the development of a methodology to certify the Multics and SFEP kernels. These efforts were performed by Stanford Research Institute (SRI) as a subcontractor to Honeywell.

For the preliminary Multics interface, SRI formulated statements of the *-property, and the simple security property in terms of the assertion language (SPECIAL) and the function of the interface. These statements, which are interpretations of the Bell and LaPadula properties, are now quite concise and precise. In addition, SRI formulated rules for the constructs of SPECIAL which, if satisfied by the specifications, guarantee the correctness of the security statements. Sample proofs of the specifications have been carried out manually. SRI will further investigate a software tool which will support these proofs.

SRI issued two technical coordination letters, TCL-14 and TCL-15 and the previously proposed "Multics Security Kernel Certification Plan, was revised during this reporting period.

7.0 CONFIGURATION MANAGEMENT

During this reporting period, a Configuration Management Plan for the Prototype Secure Multics was developed. This plan considers the configuration requirements for both the hardware and software items of the system, and specifically defines the procedures and program group interfaces that are necessary to control the three major configuration areas of Identification, Control and Accounting/Audit.

This plan identifies the program specifications and engineering documentation that establish the functional, allocated and product baselines and shows how these configuration milestones are sequenced to support the major program design reviews and configuration audits. As described in this plan, design reviews and audits will be held for the Secure Front-End Processor Hardware, Secure Front-End Processor Software and the Prototype Secure Multics.

During the next period, the initial Configuration Item Development Record will be prepared for each configuration item. This document is used to plan the specification completion dates in relation to the Preliminary and Critical Design Reviews and estimate the dates for the Functional and Physical Configuration Audits.

REFERENCES

1. James P. Anderson, Computer Security Technology Planning Study, ESD-TR-73-51, October 1972.

2. R. Schell, P. Downey, G. Popek, Preliminary Notes on the Design of a Secure Military Computer System, MCI-73-1, January, 1972.

3. D. E. Bell, L. J. LaPadula, Secure Computer Systems, ESD-TR-73-278, The MITRE Corporation, Bedford, Mass.

4. W. R. Price, Implications of a Virtual Memory Mechanism for Implementing Protection in a Family of Operating Systems, PhD Thesis, Carnegie-Mellon University, June, 1973.

5. E. L. Burke, Synthesis of a Software Security System, MTP-154, The MITRE Corporation, Bedford, Mass.

6. D. E. Bell, E. L. Burke, A Software Validation Technique for Certification: The Method, ESD-TR-75-54, The MITRE Corporation, Bedford, Mass, December, 1973.

7. L. Robinson, P. G. Neumann, K. N. Levitt, A. Saxena, "On Attaining Reliable Software for a Secure Operating System", 1975 International Conference on Reliable Software, Los Angeles, Ca, April, 1975.

8. W. L. Schiller, Design of a Security Kernel for the PDP-11/45, The MITRE Corporation, Bedford, Mass, December, 1973.

9. W. L. Schiller, The Design and Specification of a Security Kernel for the PDP-11/45, The MITRE Corporation, Bedford, Mass, March, 1975.

10. L. Smith, Architectures for Secure Computing Systems, ESD-TR-75-51, The MITRE Corporation, Bedford, Mass, June, 1974.

11. J. C. Whitmore, A. Bensoussan, P. A. Green, A. M. Kobziar, J. A. Stern, Design for Multics Security Enhancements, ESD-TR-74-176, Honeywell Information Systems, Inc., 1974.

12. Access Isolation Mechanism Pre-Release Documentation, Honeywell Information Systems, Inc., McLean, Virginia, August, 1975.

13. External I/O in a Secure Multics, CDRL A024.

14. Multics Security Kernel Certification Plan, dated 30 July 1976, CDRL A020 (Draft).

15. A. Huber, "A Multi-process Design of a Paging System," S.M. and E.E. Thesis, Department of Electrical Engineering and Computer Science, MIT, May 1976.

16. D. Reed, "Process Multiplexing in a Layered Operating System," S.M. Thesis, Department of Electrical Engineering and Computer Science, MIT, June 1976.

17. W. Montgomery, "A Secure and Flexible Model of Process Initiation for a Computer Utility," S.M. and E.E. Thesis, Department of Electrical Engineering and Computer Science, MIT, June 1976.

17. Secure Communications Processor Architectural Study, Final Report, Contract F19628-74-C-0205, Draft, 25 August 1975.

19. Secure Communications Processor Selection Trade Study, dated 31 March 1976, CDRL A015 (Draft).

20. L. Smith, "Architectures for Secure Computing Systems," MTR-2772, dated April 2075.

21. S.O.W. for Secure Multics Design, Development and Certification, Contract F19628-74-C-0193, dated 22 June 1975.

22. Security Protection Module Specification, DS 34025843, dated 31 January 1976, CDRL A021 (Draft).

23. Security Protection Module Specification, DS 34025843 dated 25 June 1976,CDRL A021 (Draft).

24. Probabilistic Measures of Compromise, dated 31 January, 1976, CDRL A019 (Draft).

APPENDIX A

RUGGEDIZED LEVEL 6

COMPUTER DEVELOPMENT PROGRAM

I. INTRODUCTION

The objective of the Honeywell funded RNML development program is to ruggedize the commercial (HIS) Level 6 minicomputers for military and non-benign commercial environmental applications. The ruggedized computer is the hardware base for the development of the SCOMP.

In order to contain costs within specified guidelines, the computer design is oriented primarily toward ground, ground-mobile, and transport aircraft applications. Primary design effort is directed towards the following:

1. New chassis structure
2. Circuit card stiffening
3. Pin and socket circuit card connector changes for the bus interface
4. Power supply mounting
5. New control panel structure
6. Cost reduction

II. DESIGN APPROACH

Chassis Design

An important consideration for militarized or ruggedized equipment is the ability of the mechanical structure to withstand vibration and shock levels typical of a military environment. Therefore, the Level 6 computer ruggedization program involves the use of a precision investment casting as the housing for the circuit boards and power supplies. The relatively large size of the chassis container coupled with intricate external rib patterns and good dimensional accuracy capability of the precision investment casting process are combined to produce a design that is both cost-effective and sufficiently rugged to meet a variety of military service environments.

Figure 1 is a photograph of the assembled Level 6 ruggedized computer.

Figure 2 shows several important internal features of the ruggedized Level 6 computer chassis such as the RFI shielded bulkhead between the logic and power supply sections of the unit. Also shown in the photograph of Figure 2 is the inside of the

front control panel depicting the EMI shielded air intake openings and the continuous RFI gasket used to seal the panel to chassis interface.

The rear view of the assembled RL6 unit, shown in Figure 3, indicates the locations and type of electrical connectors used to interface this unit as well as the shielded ventilating air exhaust panel.

Board Stiffener Concept

Figure 4 shows a comparison of a typical Level 6 computer board with and without stiffeners. Since the SFEP application does not involve exposure to vibration or severe shock environments, the RNML board stiffener design will be modified for the SFEP application; i.e., only a limited number of stiffiners will be used to control board deflections during board handling, installation, and removal operations. For the SFEP application, the longitudinal stiffener members, as shown in Figure 4, will be deleted to simplify production and reduce costs.

FIGURE 1

FIGURE 2
RUGGEDIZED LEVEL 6 CHASSIS DETAILS

FIGURE 3
REAR VIEW OF RLG ASSEMBLE

FIGURE 4
CPU BOARD WITH/WITHOUT STIFFENERS

Circuit Board Connectors

The Level 6 computer ruggedization program incorporates an improved plug-in connector for electrical interfacing of the motherboards. Because of the large number of electrical connections associated with the computer backpanel, the electrical connector used in this application represents a critical component from a reliability standpoint. Accordingly, the basic minicomputer design has been modified to provide a connector mechanization with superior dynamic capability and greater long-term reliability than the existing card-edge system. The proven "blade and fork" connector design has been used successfully on several Honeywell Aerospace programs.

The connector selected for the RL6 application will provide improved reliability for both static (ground/laboratory) and dynamic (aircraft/mobile) environments by reducing problems associated with corrosion, oxidation, humidity and dust typically encountered in actual service conditions.

A significant feature of the connector mechanization is that this approach permits use of the existing multilayer backpanel design (without change). Positive alignment of the mating connector pins during board installation is accomplished by three stainless steel dowel pins. Additionally, these dowel pins and their mating nylon bushings will provide adequate support for the board and stiffener member to ensure that the connector contacts are not stressed under mechanical loads during potential shock conditions associated with bench handling/transportation environments.

One design option of the Level 6 provides for use of edge board connectors with auxiliary board stiffeners/alignment features as shown in the photograph of Figure 5.

- Power Supply

In order to meet the overall EMC and TEMPEST requirements, special attention is required in the power supply area. The basic approach to meeting these requirements is to mechanically isolate the power supply in a separate compartment and to electrically decouple the unwanted energy from the external prime power input lines and from each internal DC power source. Special attention was given to grounding, shielding, and power transmission line impedances.

The existing power supply was modified to incorporate a structurally different method of mechanical attachment into the main RNML chassis as shown in Figure 6.

Figure 6 shows the power supply mounting container (left foreground) used to interface the power supply into the RNML main chassis. The power supply shown in the right portion of Figure 6 is the commercial version used in the Level 6 computer.

Control Panel

Design modifications have been made to ensure that the control panel satisfies the vibration/shock and EMC/TEMPEST protection requirements. The design approach for the control panel involves the use of a separate precision investment casting as the primary structural element for the panel. Suitable RFI sealing gaskets and screened-shielded air inlet openings will be incorporated to provide specification performance for the EMC/TEMPEST requirements. Details of the assembled control panel are shown in Figure 7.

Environmental Design Capability

The details and specifications of Table 1 establish the intended design capability for the RNML equipment. A more detailed description of the RNML is provided in Honeywell Document No. DS-BG8249A1 "Ruggedized Level 6 Computer (RL6)", December 15, 1975.

FIGURE 5
RUGGEDIZED LEVEL 6 COMPUTER BUSS BOARDS
AND INTERFACE WIRING

FIGURE 6
LEVEL 6 POWER SUPPLIES
RUGGEDIZED/COMMERCIAL VMT COMPARISON

FIGURE 7
RUGGEDIZED LEVEL 6 CONTROL RNML

TABLE 1

ENVIRONMENTAL DESIGN

Environment	Specification	Remarks
Operating Temperature	MIL-E-4158	0 deg. C to +52 deg. C
	MIL-E-16400	0 deg. C to +52 deg. C
Non-Operating Temperature	MIL-E-4158	-62 deg. C to +52 deg. C
	MIL-E-16400	-62 deg. C to +75 deg. C
	MIL-E-5400	-62 deg. C to +85 deg. C
Vibration	MIL-E-4158	2G Peak, Hard-Mounted
	MIL-E-16400	
	MIL-E-5400	(Curve 2G Peak, Hard-Mounted IIA)
	MIL-E-5400	(Curve 10G Peak, with IA Isolators)
Shock	MIL-E-4158	
	MIL-E-5400	15G's, 11 ms
	MIL-E-16400	With Isolators
Humidity	MIL-E-4158	
	MIL-E-16400	
	MIL-E-5400	
Altitude (Operating)	MIL-E-4158	0 - 8,000 Feet
Salt Spray Test	MIL-E-16400	
Electromagnetic Compatibility	MIL-STD-461	
TEMPEST	NACSEM 5100	As modified by DCA Circular 370-D195-2

III. RL6 QUALIFICATION TESTS

The RNML will be subjected to inspection and proof tests as defined in Honeywell's Qualification Test Plan dated January 31, 1976, to verify that the RNML and its components meet the intended specification requirements. The RL6 Qualification Unit will be fabricated and inspected to engineering released drawings.

A more detailed description of the Qualification Tests to be performed is provided in Honeywell's "Qualification Test Plan", QTP-BG8249A1, January 31, 1976.

IV. RL6 DEVELOPMENT STATUS

All engineering drawings for Qualification Unit fabrication have been formally released.

Qualification Unit has been fabricated and assembled and is now in the process of electrical checkout prior to start of qualification testing.

V. PLANS

A. After the ruggedized Level 6 computer is operational, the following preliminary engineering tests will be performed prior to the start of the formal qualification test program:

1. A minimum 100 hour "burn-in" of electrical components at room temperature ambient.

2. Workmanship vibration testing at 1G sinusoidal, 10 - 2000 Hz at 1 octave per minute.

3. Complete functional checkout of unit.

B. Qualification test sequence/steps will be as follows:

1. High temperature

2. Low temperature

3. Altitude

4. EMC

5. Vibration

6. Shock (Half-Sine)

7. Humidity

8. Shock (Navy Hammer Drop, MIL-S-901)

9. Retrofit test unit with blade and fork connectors

10. Vibration with blade and fork connectors

11. Shock (half-sine) with blade and fork connectors

12. Humidity with blade and fork connectors

13. Shock (Navy Hammer Drop, MIL-S-901) with blade and fork connectors

APPENDIX B

DOCUMENTATION

This Appendix lists the major documentation that was produced during the past six months. Due to the imminent availability of several significant reports and specifications in July 1976, additional documentation currently in preparation is also listed.

Other internal notes and working documents were also developed during the last six months but are not included in the following lists.

Technical Coordination Letters (TCL's)

Honeywell's Technical Coordination Letters provide a series of technical notes to document important technical information and issues from investigation, technical reviews and working meetings. The TCL's issued during this period were:

TCL No.	Date	Title
TCL-11	6 Jan 1976	Meeting Minutes - SFEP Software Technical Interchange Meeting - 9 Dec 1975
TCL-12	19 Jan 1976	Meeting Minutes - SRI Activities - Technical Interchange Meeting 15 Jan 1976
TCL-13	27 Jan 1976	PL/I as a System Programming Language for a Certifiable Multics
TCL-14	27 Jan 1976	Preliminary Modularization of of Multics
TCL-15	31 Jan 1976	Initial Description of Multics I/O
TCL-16	30 Jan 1976	Preliminary Test and Evaluation Plan
TCL-17	26 Mar 1976	A Technical Note on Discretionary Access Control
TCL-18	26 Mar 1976	Progress Report - Stanford Research Institute
TCL-19	1 Apr 1976	SFEP Design Review Package for Technical Interchange Meeting on 6 and 7 April
TCL-20	20 Apr 1976	Meeting Minutes - SFEP Technical Interchange Meeting - Apr 6 and 7, 1976
TCL-21	12 May 1976	Summary of the April 29, 1976 Technical Meeting - Multics
TCL-22	4 Jun 1976	Technical Coordination Meeting - CDRL Item A014, Security and Integrity Procedures

Contract Data Items

During this reporting period, numerous technical reports and specifications were submitted to the Air Force for review and approval. The following is a list of these Data Items (CDRL's) and their respective status. Also included are those CDRL's that will be submitted to the Air Force in July 1976.

CDRL No.	Date	Title
A001	15 Apr 1976	Quarterly Report for the Period, January 1976 - March 1976
	20 Jul 1976	Quarterly Report for the Period, April 1976 - June 1976
A002	1 Mar 1976	Monthly Report for January 1976
	10 Mar 1976	Monthly Report for February 1976
	21 May 1976	Monthly Report for April 1976
	11 Jun 1976	Monthly Report for May 1976
A004	27 Jan 1976	Interim Report (Final)
A005	31 Jan 1976	Final Report (Revised Draft)
A006	31 Mar 1976	Multics Security Integration Requirements (Revised Draft)
A007	16 Apr 1976	Effects of Producing a Multics Security Kernel (Final)
A008	31 Jan 1976	Multics Security Kernel Specification (Preliminary Draft)
	30 Jul 1976	Multics Security Kernel Specification (Revised Draft)
A013	31 Jan 1976	Prototype Secure Multics Specification (Preliminary Draft)
A014	30 Jul 1976	Security and Integrity Procedures (Revised Draft)
A015	31 Mar 1976	Secure Communications Processor Selection Trade Study (Revised Draft)
A016	12 Jul 1976	SFEP Processor Manual (Revised Draft)

A017	31 Jan 1976	6000/Series 60 Interface Unit Specification (Draft)
A018	31 Jan 1976	SFEP Kernel Development Specification (Preliminary Draft)
	29 Feb 1976	SFEP Kernel Development Specification (Revised Draft)
A019	31 Jan 1976	Probabilistic Measure of Compromise (Draft)
A020	27 Jan 1976	Certification Methodology Approach Plan (Draft)
	30 Jul 1976	Certification Methodology Approach Plan (Revised Draft)
A021	31 Jan 1976	Security Protection Module Specification (Draft)
A022	31 Jan 1976	Semi-Annual Progress Report - July 1975 - Dec. 1975 (Draft)
A025	12 Jul 1976	SCOMP Hardware Test Plan (Draft)
A026	30 Jul 1976	Configuration Management Plan (Draft)

APPENDIX C

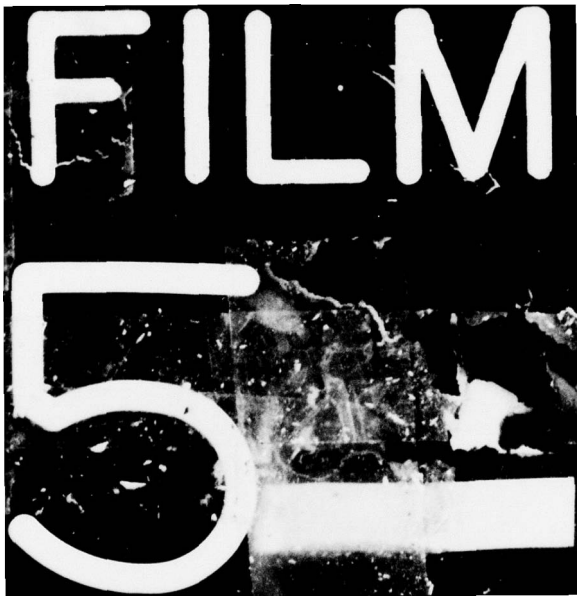
AIR FORCE ELECTRONIC SYSTEM DIVISION COMMENTS

APPENDIX C

Comments on "Semi-Annual Progress Report" Dated 6 August 1976 (CDRL Item A022)

Para 2.4 Specific quantitative information on the reduction in size and complexity would be helpful.

Para 4.0 This paragraph is somewhat misleading in several respects. (1) The paragraph does not mention security kernels and, instead, talks of securing the operating system and verifying its correctness. (2) The *-property addresses the "Trojan Horse" threat. (3) Security procedures will insure that no trap doors exist. (4) The verification must show that the kernel is tamperproof which precludes the subsequent insertion of a trap door by a malicious user.


FILM